The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

ENCAPSULATION SERVICE

DRAFT RECOMMENDED STANDARD

CCSDS 133.1-P-1.1

PINK SHEETS

March 2008

CONTENTS (continued)

Figure Page

Table

this document, and are transferred using the Packet Service of a Space Data Link Protocol. When Space Packets are used for encapsulation, some rules on their format (described in subsection 4.1) shall be applied. Management shall establish which Space Data Link Protocol is to be used to transfer encapsulated data units.

2.2 FEATURES OF ENCAPSULATION SERVICE

The Encapsulation Service transfers a sequence of variable-length, delimited, octet-aligned data units with a Space Data Link Protocol over a space link. A user of this service is a protocol entity that sends or receives protocol data units that do not have an authorized PVN. The format and content of data units transferred with this service are unknown to the service provider.

A data unit supplied by the service user is encapsulated unchanged into a Space Packet or an Encapsulation Packet and no more than one data unit is encapsulated into a single packet. An implementation is conformant if it implements either the Space Packet or the Encapsulation Packet; i.e., both are not required.

The service permits a data unit to be of any length which is an integral number of octets, and which is subject to the maximum and minimum sizes established by the project organization. Although the maximum length of a data unit that can be accommodated by an encapsulating packet is 65,536 octets (if the Space Packet is used) or 4,294,967,~~291~~287 octets (if the Encapsulation Packet is used), individual project organizations may establish the maximum and minimum sizes for the encapsulated data unit.

The point at which an instance of this service is provided to a user is called a Service Access Point (SAP) [6]. Data units submitted to a SAP are processed in the order of submission. No processing order is maintained for data units submitted to different SAPs.

NOTE – Implementations may be required to perform flow control at an SAP between the service user and the service provider. However, CCSDS does not recommend a scheme for flow control between the user and the provider.

Features of the Encapsulation Service are as follows:

a) Unidirectional (one way) service: one end of a connection can send, but not receive, data through the space link, while the other end can receive, but not send, data through the space link.

b) Asynchronous service: There are no timing relationships between the transfer of data units supplied by the user and any data transmission mechanism within the Data Link Layer. The user may request data transfer at any time, but there may be restrictions imposed by the service provider on the data generation rate.

c) Unconfirmed service: the sending user does not receive confirmation from the receiving end indicating that data has been received.

3 SERVICE DEFINITION

3.1 OVERVIEW

This section provides service definition in the form of primitives, which present an abstract model of the logical exchange of data and control information between the service provider and the service user. The definitions of primitives are independent of specific implementation approaches.

The parameters of the primitives are specified in an abstract sense and specify the information to be made available to the user of the primitive. The way in which a specific implementation makes this information available is not constrained by this specification. In addition to the parameters specified in this section, an implementation may provide other parameters to the service user (e.g., parameters for controlling the service, monitoring performance, facilitating diagnosis, and so on).

3.2 SERVICE PARAMETERS

NOTE – The parameters used by the Encapsulation Service primitives are described in subsections 3.2.1 through 3.2.5.

3.2.1 DATA UNIT

The parameter Data Unit is the service data unit transferred by the Encapsulation Service, and it shall be a delimited, octet-aligned data unit.

Although the maximum length of a data unit that can be accommodated in an encapsulating packet is 65,536 octets (if the Space Packet is used) or 4,294,967,~~291~~287 octets (if the Encapsulation Packet is used), individual project organizations may establish the maximum and minimum sizes for the encapsulated data unit.

3.2.2 GVCID

The Global Virtual Channel Identifier (GVCID) is part of the SAP address of the Encapsulation Service, and it shall indicate the Virtual Channel of the underlying Space Data Link Protocol through which the Data Unit is to be transferred.

3.2.3 PVN

The Packet Version Number (PVN) is part of the SAP address of the Encapsulation Service, and it shall indicate whether the Space Packet or the Encapsulation Packet is to be used for encapsulating the Data Unit. The value of the PVN shall be either 1 (if the Space Packet is used) or 8 (if the Encapsulation Packet is used).

4 DATA UNITS AND PROCEDURES

4.1 SPACE PACKET

The format of the Space Packet is defined in reference [5].

The following rules on the format of the Space Packet shall be applied when the Space Packet is used for encapsulating data units supplied by the service user:

a) the Packet Secondary Header shall be absent, and the value of the Secondary Header Flag shall be '0';

b) the Application Process Identifier (APID) shall be chosen from one of the reserved APIDs in the 2040 to 2044 range documented in reference [8];

c) the value of the Sequence Flags shall be '11';

d) the Packet Sequence Count shall always be used instead of a Packet Name;

e) one data unit supplied by the service user shall be placed in the User Data ~~F~~field of a Space Packet.

4.2 ENCAPSULATION PACKET

4.2.1 GENERAL

An Encapsulation Packet shall encompass the major fields, positioned contiguously, in the following sequence:

a) Packet Header (1 to 8 octets, mandatory);

b) Encapsulated Data Unit (from 0 to 4,294,967,~~291~~287 octets, optional).

An Encapsulation Packet shall consist of at least 1 and at most 4,294,967,~~296~~295 octets.

NOTE – The maximum Encapsulation Packet length allowed by a particular spacecraft or ground implementation may be less than the maximum specified here.

The structural components of the Encapsulation Packet are shown in figure 4-1.

NOTE – The definition of the Encapsulation Packet Header, below, differs from and is incompatible with the initial definition contained in references [B2]-[B4]. The definition below replaces the initial definition and renders it obsolete.

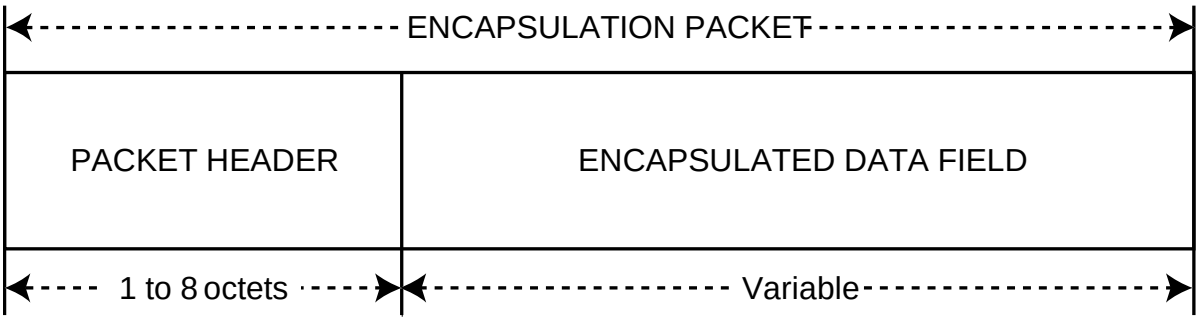

Figure 4-1: Encapsulation Packet Structural Components

4.2.2 ENCAPSULATION PACKET HEADER

4.2.2.1 General

The Packet Header is mandatory and it shall have a length of one, two, four, or eight octets. The Packet Header shall consist of three, four, ~~five~~six, or ~~six~~seven fields, positioned contiguously, in the following sequence:

a) Packet Version Number (3 bits, mandatory);

b) Protocol ID (3 bits, mandatory);

c) Length of Length (2 bits, mandatory);

d) User Defined ~~F~~field (4 bits; mandatory in 4- and 8-octet headers; not used in 1- and 2-octet headers—see figure 4-2);

e) Protocol ID Extension ~~F~~field (4 bits; mandatory in 4- and 8-octet headers; not used in 1- and 2-octet headers—see figure 4-2);

f) CCSDS Defined ~~F~~field (2 octets; mandatory in 8-octet headers; not used in 1-, 2-, and 4-octet headers—see figure 4-2);

g) Packet Length (~~0,~~ 1, 2, or 4 octets; mandatory in 2-, 4-, and 8-octet headers; not used in 1-octet headers—see figure 4-2).

The format of the Packet Header is shown in figure 4-2.

← ENCAPSULATION PACKET HEADER →						
PACKET VERSION NUMBER 3 bits	PROTOCOL ID 3 bits	LENGTH OF LENGTH 2 bits	USER DEFINED FIELD 0 or 4 bits	PROTOCOL ID EXTENSION 0 or 4 bits	CCSDS DEFINED FIELD 0 or 2 octets	PACKET LENGTH 0 to 4 octets
'111'	'XXX'	'00'	0 bits	0 bits	0 octets	0 octets
'111'	'XXX'	'01'	0 bits	0 bits	0 octets	1 octet
'111'	'XXX'	'10'	4 bits	4 bits	0 octets	2 octets
'111'	'XXX'	'11'	4 bits	4 bits	2 octets	4 octets

Figure 4-2: Packet Header

4.2.2.2 Packet Version Number

Bits 0-2 of the Packet Header shall contain the (Binary Encoded) Packet Version Number.

This 3-bit field shall identify the data unit as an Encapsulation Packet defined by this subsection; it shall be set to '111'.

NOTE – The Version Number is used to reserve the possibility of introducing other packet structures. This subsection defines 'Encapsulation Packet (Version 8 CCSDS Packet)' whose Binary Encoded Version Number is '111'.

4.2.2.3 Protocol ID

Bits 3-5 of the Packet Header shall contain the Protocol ID.

The Protocol ID shall be used to identify the protocol whose data unit~~s are being~~ is encapsulated within the Encapsulation Packet.

The Protocol IDs ~~allowed~~recognized by CCSDS for the Encapsulation Packet shall be registered in reference [8].

~~The value '110' in the Protocol ID field shall signal that the 4-bit Protocol ID Extension field is used for protocol identification.~~

~~NOTE – The protocol ID '111' is used for sending mission-specific, privately defined data (i.e., not data units of a protocol to which a Protocol ID is assigned in reference [8]) with the Encapsulation Packet).~~

NOTES

1 The value ‘000’ in the Protocol ID field signals that the packet is an Encapsulation Idle Packet. Encapsulation Idle Packets may be used to fill space in a fixed-length Transfer Frame used in references [1] and [3].

2 The value ‘110’ in the Protocol ID field signals that the 4-bit Protocol ID Extension field is used for protocol identification.

3 The value ‘111’ in the Protocol ID field signals that the Encapsulated Data field contains mission-specific, privately defined data.

4.2.2.4 Length of Length

Bits 6-7 of the Packet Header shall contain the Length of Length field.

The Length of Length shall ~~be used to specify~~define the length of the Packet Length field.

The value~~s~~ of this field shall be interpreted as shown in table 4-1.

Table 4-1: Interpretation of Length of Length Field

Value of ‘Length of Length’ Field (binary)	Length of ‘Packet Length’ Field
00	0
01	1 octet
10	2 octets
11	4 octets

~~The value ‘00’ of the Length of Length field shall only be used for Encapsulation Packets with Protocol ID ‘000’ (i.e., Fill Packets).~~

~~NOTE — When the value of the Length of Length field is ‘00’, there is no Packet Length field in the Packet Header, and there is no Encapsulated Data Unit field in the Encapsulated Packet. Therefore the length of the Encapsulated Packet is one octet. *This one-octet Encapsulation Packet can thus also be used as a single, self-identified octet of fill which may be cascaded to provide any number of octets to fill a fixed-length Transfer Frame used in references [1] and [2].*~~

If the Length of Length field has the value ‘00’ then the Protocol ID field shall have the value ‘000’, indicating that the packet is an Encapsulation Idle Packet.

NOTE – If the Length of Length field has the value ‘00’, then the Packet Length field and the Encapsulated Data Unit field are both absent from the packet. In this case, the length of the Encapsulation Packet is one octet.

4.2.2.5 User Defined Field

If present, the User Defined field shall be four bits in length and shall follow, without gap, the Length of Length field.

The User Defined field shall be used for sending mission-specific, privately defined header data.

4.2.2.6 Protocol ID Extension

If present, the Protocol ID Extension field shall be four bits in length and shall follow, without gap, the User Defined ~~F~~field.

~~The~~If the Protocol ID field contains the value '110', then the Protocol ID Extension field shall be used to identify the protocol whose data unit~~s are being~~ is encapsulated within the Encapsulation Packet. If the Protocol ID field does not contain the value '110', then the Protocol ID Extension field is by convention set to 'all zeros'.

The extended protocol IDs ~~allowed~~recognized by CCSDS for the Encapsulation Packet shall be registered in reference [8].

4.2.2.7 CCSDS Defined Field

If present, the CCSDS Defined field shall be two octets in length and shall follow, without gap, the ~~User Defined~~Protocol ID Extension field.

The CCSDS Defined field is reserved for future use by CCSDS and is by convention set to 'all zeros'.

4.2.2.8 Packet Length

If present, the Packet Length field shall be the final field in the Encapsulation Packet ~~header~~Header (see figure 4-2).

If the value of the Length of Length field is '00', the Packet Length field shall be absent. Otherwise, the Packet Length field shall contain a binary number corresponding to the total length of the Encapsulation Packet (in octets), including the Packet Header. See table 4-2.

NOTES

1 Although unlikely to be used in space, a 4-octet Packet Length field permits accommodating IPv6 (reference [B6]) 'Jumbograms' up to 4,294,967,~~291~~287 (=2^{32}-5) octets in length.

2 If the Packet Length field is absent (i.e., the value of the Length of Length field is '00'), then the length of the Encapsulation Packet is one octet.

Table 4-2: Encapsulation Packet Lengths, Depending on the Length of the Packet Header

Length (*) of Packet Header	Number of octets in Packet Length Field	Minimum packet length	Maximum packet length	Minimum length of Encapsulated Data Field	Maximum length of Encapsulated Data Field (**)
1	Packet Length Field is absent	1	1	Encapsulated Data Field is absent	
2	1	2	255	0	253
4	2	4	65 535	0	65 531
8	4	8	4 294 967 295	0	4 294 967 287

(*) All lengths are given in octets.
(**) An implementation may establish a lower value for the maximum length of the Encapsulated Data field. The length of the Encapsulated Data Unit is the same as the length of the Encapsulated Data field.

4.2.3 ENCAPSULATED DATA FIELD

If present, the Encapsulated Data ~~F~~field shall follow, without gap, the Packet Length field. It shall consist of an integral number of octets.

~~If the value of the Length of Length field is '00', the Encapsulated Data Field shall be absent. Otherwise, it shall contain a data unit, supplied by the service user, consisting of an integral number of octets.~~

The Encapsulated Data field shall contain the protocol data as indicated by the Protocol ID field, and by the Protocol ID Extension field if present.

If the Protocol ID field contains the value '000', then the Encapsulated Data field shall contain idle data.

Under the following conditions, the Encapsulated Data field shall be absent:

a) when the value of the Length of Length field is '00', or

b) when the value of the Length of Length field is other than '00', and the packet length as indicated by the Packet Length field is equal to the length of the Encapsulation Packet Header.

If the Encapsulated Data field is absent then the Protocol ID field shall contain the value '000'.

NOTES

1 When the Encapsulation Packet is generated as a result of an ENCAPSULATION.request to the Encapsulation Service, then the Encapsulated Data field contains the Data Unit supplied by the service user.

2 If the Protocol ID field contains the value '000', then the packet is an Encapsulation Idle Packet. Encapsulation Idle Packets may be used as Idle Packets by the TM Space Data Link Protocol (reference [1]) and by the AOS Space Data Link Protocol (reference [3]). These protocols generate Idle Packets when needed to maintain synchronization of the data transport processes.

3 CCSDS does not specify the pattern of idle data.

4.3 PROCEDURES AT THE SENDING END

NOTE – This subsection describes procedures for providing the Encapsulation Service at the sending end (see figure 4-3). The procedures described here are defined in an abstract sense and are not intended to imply any particular implementation approach of the service.

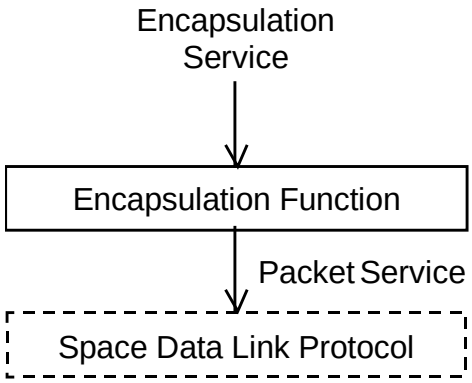

Figure 4-3: Internal Organization of Encapsulation Service (Sending End)

The Encapsulation Function (see figure 4-1) shall be used to encapsulate data units supplied by the service users. There is an instance of the Encapsulation Function for each packet structure used for encapsulation (i.e., one instance for Space Packets and one for Encapsulation Packets).

The Encapsulation Function receives data units from the service users. Any data unit that violates the limits of the size shall be rejected. Each valid data unit shall be encapsulated, either in a Space Packet or in an Encapsulation Packet, and passed to the Packet Service of the underlying Space Data Link Protocol.